DATA ANALYSIS ON E- STORE

Project Author: Neha Kunwar Solanki

OBJECTIVE:

In this project, we are working on E-commerce Data to get some information so that the information can be used for analytical purpose and decision making, useful for Maximizing Business Profit. Huge data sets will help Organizations to address potential customers in a meaningful way.

Dataset information that could be used for future decisions, improve customer engagement so that newly launch product information can be shared to them.

SAMPLE DATA SETS FOR ANALYSIS:

1. **Customer:**

Customer_id	First name	last name	age	Address

2. Transaction:

Date	uid	amount	category	product	city	state	Payment

PROJECT DESCRIPTION -

We are provided with certain use-cases to get the required data. For all the use-cases we will be using a Map-reduce approach. The Map Reduce Approach totally works on Key-Value pair as Input and Output. There will be a Driver Class, Mapper Class and a Reducer Class.

TECHNOLOGY USED:

- Apache Hadoop
- Map-Reduce Programming in java.

SOFTWARE USED:

- Eclipse IDE
- Oracle Virtual Machine
- Ubuntu
- JDK 1.7

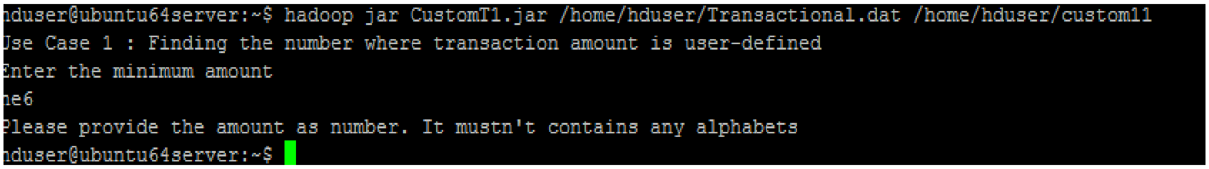
USE CASES

1. Categorization of customer based on Amount Scenario:

The system keeps track of different customer's information by their unique code. Whenever user purchases a product of a particular price or within range of amount than at time the user will provide with similar type of product within the same range.

- **Find all the transaction where amt>160**
  Validation Constraints: yes

  Output: Using Custom input



```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
ne6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$
```

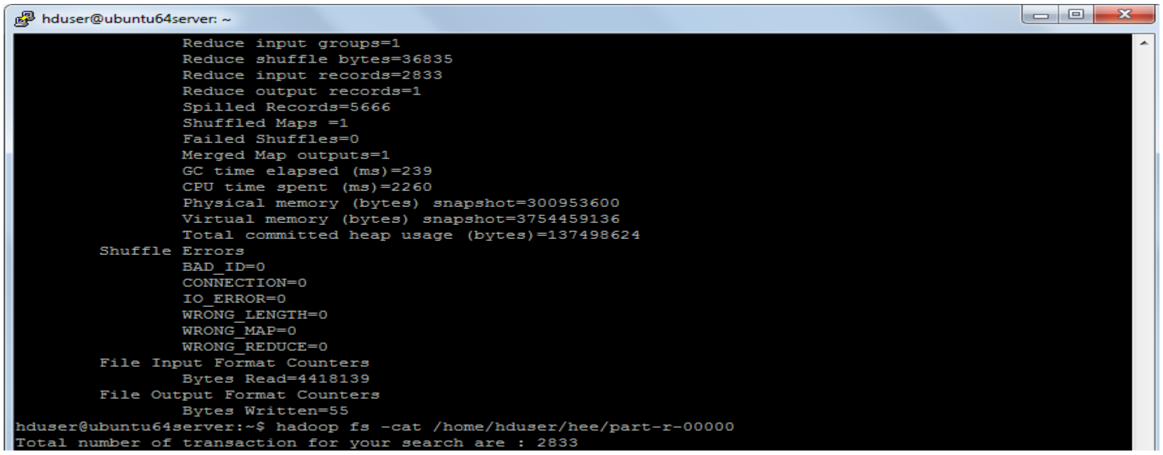
2. Customer transaction information

In this use case, we are finding all the transaction where amount is more than 170 and less than 200 paid by customer. So that we come to know about the specific amount or more than that is paid by customer for purchasing and new product and services in same rang can be shared to them.

- **Count all the transaction where amount is between 175 to 200**

  Validation Constraints: yes

  Output: Using Custom input



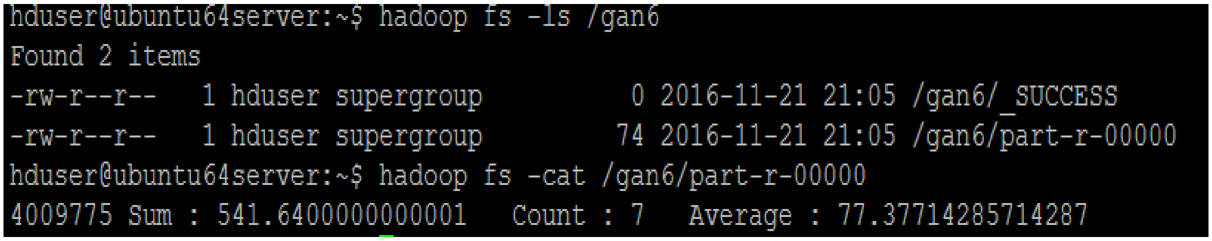
```
                Reduce input groups=1
                Reduce shuffle bytes=36835
                Reduce input records=2833
                Reduce output records=1
                Spilled Records=5666
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=239
                CPU time spent (ms)=2260
                Physical memory (bytes) snapshot=300953600
                Virtual memory (bytes) snapshot=3754459136
                Total committed heap usage (bytes)=137498624
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=4418139
        File Output Format Counters
                Bytes Written=55
hduser@ubuntu64server:~$ hadoop fs -cat /home/hduser/hee/part-r-00000
Total number of transaction for your search are : 2833
```

3. Overall Transaction counting for each user:

In this use case, we will fetch a Customer Overall Transactional report against each customer ID, Counting will be done for the no. of transactions. Also for each customer, all the transaction amount is added. Finally we will display the count and total transaction amount for every customer.

- **Calculate the total sum and total count of all the transaction for each user id**

  Output: Using Custom input& Validation Constraints



```
hduser@ubuntu64server:~$ hadoop fs -ls /gan6
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 21:05 /gan6/_SUCCESS
-rw-r--r--   1 hduser supergroup         74 2016-11-21 21:05 /gan6/part-r-00000
hduser@ubuntu64server:~$ hadoop fs -cat /gan6/part-r-00000
4009775 Sum : 541.6400000000001   Count : 7   Average : 77.37714285714287
```

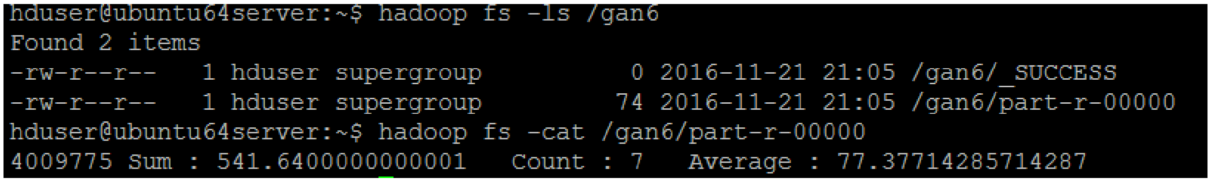
4. Calculate the average transaction value for each user id

In this case we will fetch the transaction amount data of each user and get the average of all transaction as per individual user id. Average rating for each user can be done periodically for analysis. Average transaction only decides weekly, monthly and yearly product services.

- **Calculate the total sum and total count of all the transaction for each user id**

  Validation Constraints: Yes

  Output: **<u>Using Custom input</u>**



```
hduser@ubuntu64server:~$ hadoop fs -ls /gan6
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 21:05 /gan6/_SUCCESS
-rw-r--r--   1 hduser supergroup         74 2016-11-21 21:05 /gan6/part-r-00000
hduser@ubuntu64server:~$ hadoop fs -cat /gan6/part-r-00000
4009775 Sum : 541.6400000000001   Count : 7   Average : 77.37714285714287
```

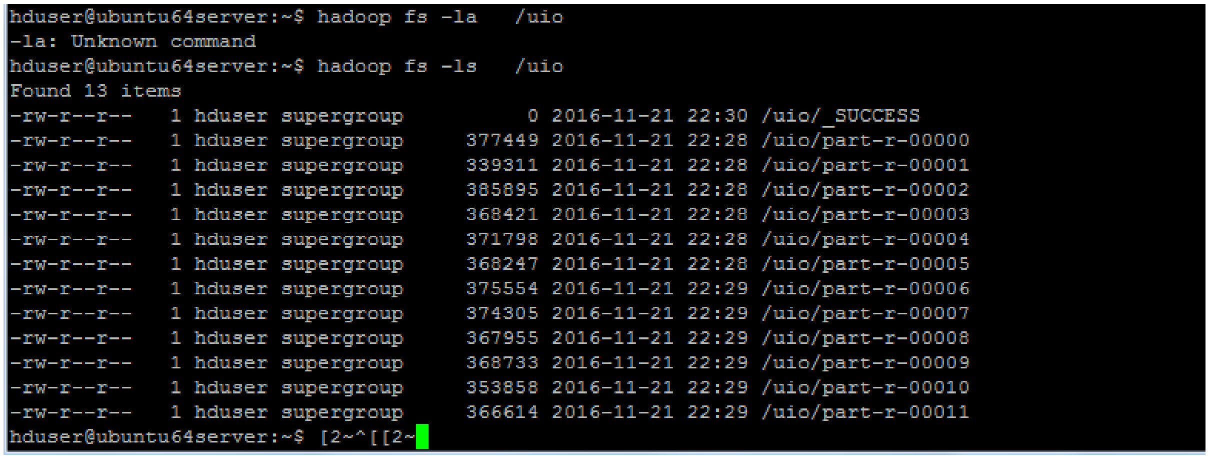
5. Division of single file into multiple files

In this use case dataset is divided into multiple sub file according to product category. As a developers, fetching data from two table and divide them according to product category. For example how many customer has used credit card for as a payment mode, How many customer took offer on products so that when season comes for discount we can inform those customer about discount offer.

- **Divide the file into 12 files, each file containing each month of data. For eg. file 1 should contain data of January txn, file 2 should contain data of feb txn.**

  Validation Constraints: Yes

  Output: <u>Using Custom input</u>



```
hduser@ubuntu64server:~$ hadoop fs -la   /uio
-la: Unknown command
hduser@ubuntu64server:~$ hadoop fs -ls   /uio
Found 13 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 22:30 /uio/_SUCCESS
-rw-r--r--   1 hduser supergroup     377449 2016-11-21 22:28 /uio/part-r-00000
-rw-r--r--   1 hduser supergroup     339311 2016-11-21 22:28 /uio/part-r-00001
-rw-r--r--   1 hduser supergroup     385895 2016-11-21 22:28 /uio/part-r-00002
-rw-r--r--   1 hduser supergroup     368421 2016-11-21 22:28 /uio/part-r-00003
-rw-r--r--   1 hduser supergroup     371798 2016-11-21 22:28 /uio/part-r-00004
-rw-r--r--   1 hduser supergroup     368247 2016-11-21 22:28 /uio/part-r-00005
-rw-r--r--   1 hduser supergroup     375554 2016-11-21 22:29 /uio/part-r-00006
-rw-r--r--   1 hduser supergroup     374305 2016-11-21 22:29 /uio/part-r-00007
-rw-r--r--   1 hduser supergroup     367955 2016-11-21 22:29 /uio/part-r-00008
-rw-r--r--   1 hduser supergroup     368733 2016-11-21 22:29 /uio/part-r-00009
-rw-r--r--   1 hduser supergroup     353858 2016-11-21 22:29 /uio/part-r-00010
-rw-r--r--   1 hduser supergroup     366614 2016-11-21 22:29 /uio/part-r-00011
hduser@ubuntu64server:~$ [2~^[[2~
```

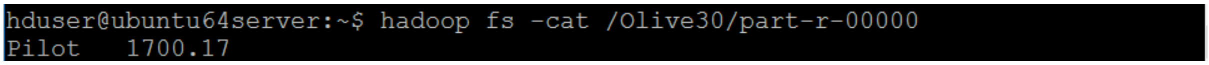
6. The profession of user who has spent the maximum amount

In this use case we are given a task to find the name of profession from customer dataset to find maximum amount. Next new products marketing starts from him by giving discount of 20% on purchasing

- **Find the profession of user who has spent the maximum amount**

  Validation Constraints: Yes

  Output: **Using Custom input**



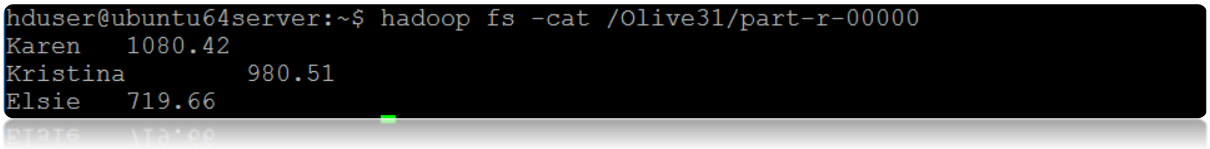
```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive30/part-r-00000
Pilot   1700.17
```

7. New Product and Services:

In this use case, we are finding three top spenders report to whom organization can offer new launching Services like yoga products, gym products, Air sports, life jackets etc.

- **Find the name of top 3 spenders.**

  Validation Constraints: Yes
  Output: Using Custom input



```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive31/part-r-00000
Karen   1080.42
Kristina        980.51
Elsie   719.66
```

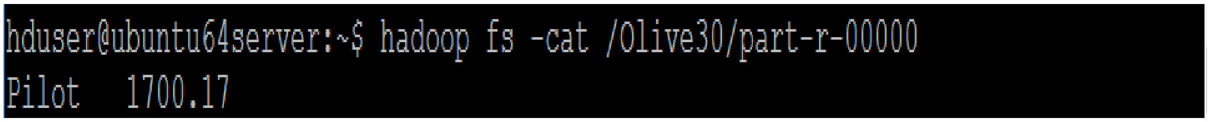
8. Retaining Customers: Customer lifetime value

In this use case, searching particular customer who has made highest transaction so that we can analyze number of unique purchase mode, average price of products, average price or orders And number of days and session leading to a transaction.

- **Find the profession of user who has spend the maximum amount**

Validation Constraints: Yes

Output: **Using Custom input**



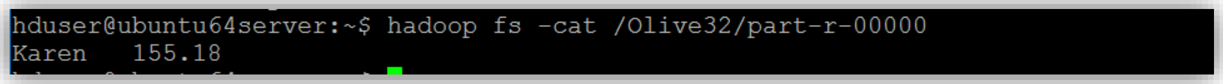
```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive30/part-r-00000
Pilot   1700.17
```

9. Special rewards: Extra point Events

In this use case, fetching all customer information in July month to give them offer like Extra point events. An extra point’s event is a great way to boost program engagement and encourage shoppers to spend points

- **Find the user who has spent the max amount in July month**

Output: **Using Custom input**



```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive32/part-r-00000
Karen   155.18
```

CONCLUSION: Analysis done on relevant very large sets of data for Statistical analysis, data mining, predictive analytics, and text mining. And we built a summary table to aggregate the detail at a monthly level transaction, a table to aggregate the detail at a year-to-date level, and a final summary table for the division level.